Control of industrial robots

Kinematic redundancy

Prof. Paolo Rocco (paolo.rocco@polimi.it)
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria
Direct kinematics of the manipulator:

\[ \mathbf{r} = f(\mathbf{q}) \]

\[ \mathbb{R}^n \rightarrow \mathbb{R}^m \]

Joint space \rightarrow Task space

A robot is kinematically redundant if: \[ n > m \]

i.e. if it has more degrees of freedom than those strictly necessary to perform a task.

Redundancy is a relative concept: the number of task variables may be less than the dimension of the operational space.

A 3 d.o.f. planar manipulator is functionally redundant if the task is specified with respect to the end-effector position only.
Why kinematic redundancy?

Kinematic redundancy can be exploited to:

- increase dexterity and manipulability
- avoid obstacles
- avoid kinematic singularities
- minimize energy consumption
- increase safety
- ....
Redundancy of the human arm

The human arm is a 7 d.o.f. system (if we do not consider the degrees of freedom of the fingers). It is thus kinematically redundant.

If we fully constrain the hand (both in terms of position and orientation), there is still one degree of freedom available (elbow or swivel angle).
Some redundant robots

KUKA LBR iiwa
Some redundant robots

ABB YuMi
Some redundant robots

Yaskawa Motoman
Some redundant robots
Some redundant robots

Rethink Robotics: BAXTER
Some redundant robots
The inverse kinematics problem is to find:

\[ q(t) : f(q(t)) = r(t), \quad \forall t \]

When the robot is redundant:

- infinite solutions exist
- the robot has \textit{self-motions}, i.e. internal motions in the joint space which do not affect the task variables

How to \textit{select a solution} of the inverse kinematics problem?
The problem is usually addressed at velocity level. Given the differential kinematics equation:

\[ \dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \]

where:

\[ \mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \]

is the (task) Jacobian, we want to solve the equation for \( \dot{\mathbf{q}} \)

Not trivial!
...because the Jacobian is a lower rectangular matrix
Consider again the differential kinematics equation:

\[ \dot{r} = J(q) \dot{q} \]

For a given configuration \( q \) the equation establishes a linear mapping from the space of joint velocities to the space of task velocities. This mapping can be characterized in terms of range and null spaces:

- **Range** \( R(J) \)
- **Null space** \( N(J) \)

If the Jacobian is full rank:

\[ \dim(R(J)) = m \]
\[ \text{rank}(J) = m \]

\[ \dim(N(J)) = n - m \]

The **null space** exists only if the robot is redundant \((n > m)\). It is the space of joint velocities that do not produce task velocities.
Consider now a matrix $P$ such that:

$$R(P) = N(J)$$

and let $\dot{q}^*$ be a solution of:

$$\dot{r} = J(q)\dot{q}$$

Then:

$$\dot{q} = \dot{q}^* + P\dot{q}_0$$

where $\dot{q}_0$ is an arbitrary vector, is also a solution.

$P$ is a projection matrix: it projects any joint velocity into the null space of the Jacobian.

Easy to prove!

$$J\dot{q} = J\dot{q}^* + JP\dot{q}_0 = J\dot{q}^* = \dot{r}$$
Null space motions

\[ \dot{q} = \dot{q}^* + P\dot{q}_0 \]

The effect of \( \dot{q}_0 \) is to generate internal motions of the manipulator, so called null space motions, which do not change the end effector position and orientation.

Three questions arise:

- How to generate a particular solution \( \dot{q}^* \)?
- How to assign an expression to the projection matrix \( P \)?
- How to generate the null space motions?
Finding a particular solution

\[ \dot{q} = \dot{q}^* + P\dot{q}_0 \]

The problem to find a particular solution can be set as the optimization of a cost function in the joint velocities, subject to the constraint given by the differential kinematics equations.

Find:

\[ \dot{q} \]

such that

\[ g(\dot{q}) \]

is minimized and:

\[ \dot{r} = J(q)\dot{q} \]

for a given \( r \).
We can take the following cost function:

\[ g(\dot{q}) = \frac{1}{2} \dot{q}^T \dot{q} = \frac{1}{2} \|\dot{q}\|^2 \]

The problem is solved with the method of the Lagrange multipliers, introducing the modified cost function:

\[ g(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T \dot{q} + \lambda^T (\dot{r} - J\dot{q}) \]

The solution has to satisfy the necessary conditions:

\[
\begin{align*}
\frac{\partial g}{\partial \dot{q}}^T &= 0 & \Rightarrow & \dot{q} - J^T \lambda = 0 \\
\frac{\partial g}{\partial \lambda}^T &= 0 & \Rightarrow & \dot{r} - J\dot{q} = 0 \\
\end{align*}
\]

\[ m \times m \text{ matrix of rank } m \text{ if } J \text{ is full rank} \]

\[ \dot{r} = JJ^T \lambda \quad \lambda = (JJ^T)^{-1}\dot{r} \]
The solution of the optimization problem is thus:

\[
\dot{q} = J^T (J J^T)^{-1} \dot{r} = J^# \dot{r}
\]

Matrix:

\[
J^# = J^T (J J^T)^{-1}
\]

is called right (Moore Penrose) pseudo inverse of the Jacobian matrix \( J \), since \( J J^# = I \).

- \( J^# \) always exists.
- if the Jacobian is not full rank, \( J^# \) can be computed numerically using Singular Value Decomposition (SVD) of matrix \( J \) (instruction \text{pinv} in Matlab)
- In this case \( J^# \) is defined as the only matrix such that:

\[
\begin{align*}
J J^# J &= J \\
J^# J J^# &= J^# \\
(J J^#)^T &= J^# \\
(J^# J)^T &= J^# J
\end{align*}
\]
If we take the alternative cost function:

\[ g(\dot{q}) = \frac{1}{2} \dot{q}^T W \dot{q} \]

where \( W \) is a \( n \times n \) symmetric positive definite matrix, we obtain the following solution:

\[ \dot{q} = J_w^# \ddot{r} \]

Matrix:

\[ J_w^# = W^{-1} J^T (JW^{-1} J^T)^{-1} \]

is called **weighted pseudo inverse** of the Jacobian matrix \( J \).

If \( W \) is diagonal it can be used to relatively weigh the joint velocities (a large \( W_i \) corresponds to small \( \dot{q}_i \)). The Moore-Penrose pseudo-inverse corresponds to the special case \( W = I \).
Null space methods

\[
\dot{q} = \dot{q}^* + P\dot{q}_0
\]

How to specify the projection matrix?

Consider the following cost function:

\[
g'(\dot{q}) = \frac{1}{2} (\dot{q} - \dot{q}_0)^T (\dot{q} - \dot{q}_0)
\]

Using again the method of the Lagrange multipliers, we can obtain:

\[
\dot{q} = J^#\dot{r} + (I_n - J^#J)\dot{q}_0
\]

\(I_n - J^#J\) is a projection matrix: it projects the velocities \(\dot{q}_0\) into the null space of the Jacobian.

\(\dot{q}_0\) is a sort of “privileged” solution.

\(\dot{q}_0\) is a sort of "privileged" solution.

null space methods

\[
\dot{q} = \dot{q}^* + P\dot{q}_0
\]
Null space methods

\[ \dot{q} = \dot{q}^* + P \ddot{q}_0 \]

How to specify vector \( \ddot{q}_0 \)?

One possible choice is the **projected gradient** method:

\[ \ddot{q}_0 = k \left( \frac{\partial U(q)}{\partial q} \right)^T \]

where \( U(q) \) is a differentiable objective function and \( k > 0 \).

We try to **locally maximize** \( U(q) \) while executing the time varying task \( r(t) \).

Notice that we will **project** the velocity \( \dot{q}_0 \) that locally maximizes the task in the null-space of the main task: the main task will not be “disturbed.”
Possible objective functions

- **Manipulability** measure (maximizes the distance from singularities):

\[ U(q) = \sqrt{\det(J(q)J^T(q))} \]

- Distance from **joint limits**:

\[ U(q) = \frac{1}{2n} \sum_{i=1}^{n} \left( \frac{q_i - \bar{q}_i}{q_iM - q_{im}} \right)^2 \]

- Distance from the **closest obstacle**:

\[ U(q) = \min_{p,o} \|p(q) - o\| \]

  - potential differentiability issues (min-max problem)
  - generic point on the robot (direct kinematics)
  - obstacle
One potential problem in the use of redundant robots is that the motion can be unpredictable:

- A closed trajectory in the task space can be mapped into an open trajectory in joint space
- Any trajectory from the same initial and final task positions may end with different joint configurations

Methods that avoid these problems are called **repeatable** or **cyclic**
Non repeatability of kinematic inversion methods is a source of problems:

- the task trajectory is usually cyclic: a repeatable joint motion is more predictable and simplifies the programming
- with a repeatable method many characteristics of the motion (joint and velocity limits, singularities) can be verified by simulating only the first cycle

The repeatability of the method has connections with the concept of holonomy of constraints in differential geometry and analytical mechanics.

Basically, in order to be repeatable, the method must enforce a holonomic constraint among the joint variables.
A way to solve the kinematic inversion problem is to add an auxiliary task:

\[
\begin{align*}
  r &= f(q) \quad \text{main task, } r \in \mathbb{R}^m \\
  y &= h(q) \quad \text{auxiliary task, } y \in \mathbb{R}^{n-m}
\end{align*}
\]

The differential kinematics for the augmented task is:

\[
\begin{align*}
  \dot{r}_A &= \begin{bmatrix} \dot{r} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} J(q) \\ \frac{\partial h}{\partial q} \end{bmatrix} \dot{q} = J_A(q) \dot{q}
\end{align*}
\]

where the augmented Jacobian \( J_A \) has been used.

A solution of the inverse kinematics is thus:

\[
\dot{q} = J_A^{-1}(q) \dot{r}_A
\]
If the auxiliary task variables $y$ are constant, the method is:

$$\dot{q} = \begin{bmatrix} J(q)^{-1} & \frac{\partial h}{\partial q} \\ \frac{\partial h}{\partial q} & 0 \end{bmatrix} \begin{bmatrix} \dot{r} \\ 0 \end{bmatrix}$$

and enforces the holonomic constraint:

$$h(q) = const$$

Therefore the extended Jacobian is a repeatable method.

- the holonomic constraints can be selected as the conditions for constrained optimality of a given objective function
- the inversion of the augmented Jacobian may introduce new singularities (algorithmic singularities)
Solutions at the acceleration level

So far we have addressed the inverse kinematic problem at velocity level. The same problem can also be addressed at **acceleration level**.

Given the differential kinematics equation:

\[ \dot{r} = J(q)\dot{q} \]

we also have:

\[ \ddot{r} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \]

Then:

\[ J(q)\ddot{q} = \ddot{r} - \dot{J}(q)\dot{q} = \ddot{x} \]

To be determined: given (at time \(t\)) known (at time \(t\))

For example, with null-space methods:

\[ \ddot{q} = J^\#(q)\ddot{x} + \left( \mathbf{I} - J^\#(q)J(q) \right)\ddot{q}_0 \]

Minimum-norm solution: projected gradient
In order to recover errors with respect to an assigned task $r_d$ due to initial mismatches, drifts, inaccuracies of the solution, etc., a **closed loop system** has to be used.

At velocity level:

$$\dot{q} = J^\#(q)[\dot{r}_d + K(r_d - r)] + \left(I_n - J^\#(q)J(q)\right)\dot{q}_0$$

where: $r = f(q)$

Similarly for the solution at acceleration level.
Human-like redundancy resolution

A.M. Zanchettin, L. Bascetta, P. Rocco

Human-like redundancy resolution for anthropomorphic industrial manipulators

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°230502
Human-like redundancy resolution